



SQL-PROJECT

on Pizza Sales





BORCELLE
RESTAURANT

HELLO!

Today, we'll embark on a data-driven journey into pizza sales. In a thriving and evolving market, understanding our sales data is essential. We'll explore how SQL can help us analyze trends, customer preferences, and performance, enabling informed decisions and strategy optimization.

Pizza is a beloved staple, and by leveraging SQL, we can gain insights to enhance our offerings and drive sales. Let's dive in and uncover the potential for pizza success!





ABOUT THE PROJECT

In this project, I analyzed pizza sales data and executed various SQL queries to extract insights related to this dataset.



DATA MODEL

Here is a brief overview of the included tables and their respective columns.

order_details	orders	pizza_types	pizzas
order_details_id	order_id	pizza_type_id	pizza_id
order_id	order_date	name	pizza_type_id
pizza_id	order_time	category	size
quantity		ingredients	price

I have performed the below SQL queries.

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.
6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the list of most ordered pizza types based on revenue for each pizza category.

1. Retrieve the total number of orders placed.

```
select count(order_id) as total_orders from orders;
```

Result Grid	
	total_orders
▶	21350

2. Calculate the total revenue generated from pizza sales.

```
select  
round(sum(order_details.quantity * pizzas.price),2) as total_revenue_genetrated  
from order_details join pizzas  
on order_details.pizza_id = pizzas.pizza_id;
```

	total_revenue_genetrated
▶	817860.05

3. Identify the highest-priced pizza.

```
select name,price  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by price desc  
limit 1;
```

| Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

```
select count(quantity) as order_count, size
from
order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
group by size
order by order_count desc
limit 1;
```

Result Grid | Filter

	order_count	size
▶	18526	L

5. List the top 5 most ordered pizza types along with their quantities.

```
select
    name, sum(quantity) as total_quantity_per_pizzatype
from
    pizza_types join pizzas
    on pizza_types.pizza_type_id= pizzas.pizza_type_id
join order_details
    on pizzas.pizza_id = order_details.pizza_id
group by name
order by total_quantity_per_pizzatype desc
limit 5;
```

	name	total_quantity_per_pizzatype
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

6. Join the necessary tables to find the total quantity of each pizza category ordered.

```
select category, sum(quantity)
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by category;
```

Result Grid | Filter Rows

	category	sum(quantity)
1	Classic	14888
2	Veggie	11649
3	Supreme	11987
4	Chicken	11050

7. Determine the distribution of orders by hour of the day.

```
select  
count(order_id) as tot_orders_placed, hour(order_time) as hours  
from orders  
group by hour(order_time);
```

Result Grid | Filter Rows:

	tot_orders_placed	hours
▶	1231	11
	2520	12
	2455	13
	1472	14
	1468	15
	1920	16
	2336	17
	2399	18
	2009	19
	1642	20
	1198	21

Result 9 ×

8. Join relevant tables to find the category-wise distribution of pizzas.

```
select * from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id;
```

	pizza_type_id	name	category	ingredients	pizza_id	pizza_type_id	size	price
▶	bbq_dkn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Pepp...	bbq_dkn_s	bbq_dkn	S	12.75
	bbq_dkn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Pepp...	bbq_dkn_m	bbq_dkn	M	16.75
	bbq_dkn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Pepp...	bbq_dkn_l	bbq_dkn	L	20.75
	cali_dkn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...	cali_dkn_s	cali_dkn	S	12.75
	cali_dkn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...	cali_dkn_m	cali_dkn	M	16.75
	cali_dkn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...	cali_dkn_l	cali_dkn	L	20.75
	dkn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...	dkn_alfredo_s	dkn_alfredo	S	12.75
	dkn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...	dkn_alfredo_m	dkn_alfredo	M	16.75
	dkn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...	dkn_alfredo_l	dkn_alfredo	L	20.75
	dkn pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...	dkn pesto_s	dkn pesto	S	12.75
	dkn pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...	dkn pesto_m	dkn pesto	M	16.75

9. Group the orders by date and calculate the average number of pizzas ordered per day.

```
select avg(total_quantity)
from
  (select order_date, sum(quantity) as total_quantity
   from
     orders join order_details
   on orders.order_id = order_details.order_id
   group by order_date)
           as ordered_quantity_perday;
```

Result Grid

	avg(total_quantity)
▶	138.4749

10. Determine the top 3 most ordered pizza types based on revenue.

```
select pizza_type_id, sum(pizzas.price * order_details.quantity) as revenue
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizza_type_id
order by revenue desc
limit 3;
```

Result Grid | Filter Rows:

	pizza_type_id	revenue
▶	thai_dkn	43434.25
	bbq_dkn	42768
	cali_dkn	41409.5

11. Calculate the percentage contribution of each pizza type to total revenue.

```
select category,
       (sum(pizzas.price * order_details.quantity)/
        (select sum(pizzas.price * order_details.quantity)
         from pizzas join order_details
          on pizzas.pizza_id = order_details.pizza_id))*100 as revenue
  from pizza_types join pizzas
    on pizzas.pizza_type_id = pizza_types.pizza_type_id
   join order_details
    on order_details.pizza_id = pizzas.pizza_id
   group by category
  order by revenue desc;
```

	category	revenue
▶	Classic	26.905960255669893
	Supreme	25.456311260099056
	Chicken	23.955137556847493
	Veggie	23.68259092738442

12. Analyze the cumulative revenue generated over time.

```
select order_date, round(sum(revenue) over (order by order_date),2) as cumulative_sales  
from  
(select order_date,  
round(sum(pizzas.price * order_details.quantity),2) as revenue  
  from pizzas join order_details  
  on pizzas.pizza_id = order_details.pizza_id  
  join orders  
  on order_details.order_id = orders.order_id  
group by order_date) as sales;
```

	order_date	cumulative_sales
▶	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.35
	2015-01-11	25862.65

13. Determine the list of most ordered pizza types based on revenue for each pizza category.

```
select name, category, revenue, rank() over (partition by category order by revenue desc)
from
(select name, category, sum(pizzas.price * order_details.quantity) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by name, category) as sales;
```

	name	category	revenue	rank() over (partition by category order by revenue desc)
▶	The Thai Chicken Pizza	Chicken	43434.25	1
	The Barbecue Chicken Pizza	Chicken	42768	2
	The California Chicken Pizza	Chicken	41409.5	3
	The Southwest Chicken Pizza	Chicken	34705.75	4
	The Chicken Alfredo Pizza	Chicken	16900.25	5
	The Chicken Pesto Pizza	Chicken	16701.75	6
	The Classic Deluxe Pizza	Classic	38180.5	1
	The Hawaiian Pizza	Classic	32273.25	2
	The Pepperoni Pizza	Classic	30161.75	3
	The Greek Pizza	Classic	28454.100000000013	4

CONTACT :

9694359189 (WhatsApp)

mamtamaximus@gmail.com

LinkedIn : @mamtachoudhary

GitHub :



THANK YOU

